Moveit![ROS topics]

Article *in* IEEE Robotics & Automation Magazine · March 2012
DOI: 10.1109/MRA.2011.2181749

Sachin Chitta
SRI International

Ioan Alexandru Sucan
Google Inc.

MoveIt!

By Sachin Chitta, Ioan Sucan, and Steve Cousins

Robots are increasingly finding applications in domains where they have to work in close proximity to humans. Industrial robotic applications are starting to examine the possibility of robots and humans as coworkers, sharing tasks and workspace. Autonomous robotic cars operating on crowded streets and freeways have to share space with pedestrians and cyclists in addition to other vehicles. Domestic robots, in particular mobile manipulation systems, will be confronted with cluttered, messy environments where obstacles exist at every corner, and people are continuously moving in and out of the workspace of the robots.

Robots working in human environments clearly have to be aware of their surroundings and must actively attempt to avoid collisions with humans and other obstacles. MoveIt! is a set of software packages integrated with the Robot Operating System (ROS) and designed specifically to provide such capabilities, especially for mobile manipulation. MoveIt! will allow robots to build up a representation of their environment using data fused from three-dimensional (3-D) and other sensors, generate motion plans that effectively and safely move the robot around in the environment, and execute the motion plans while constantly monitoring the environment for changes.

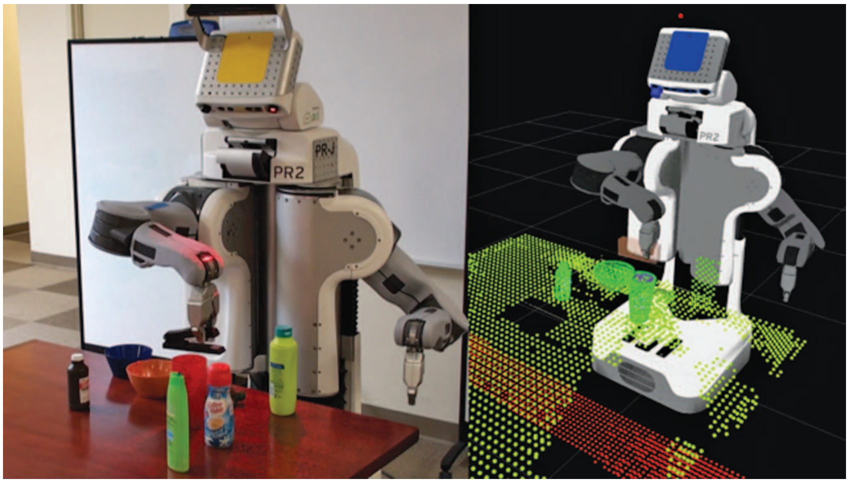
Ground Work

MoveIt! is an evolution of the arm_ navigation software in ROS. The arm_navigation packages were designed for motion planning, trajectory generation, and environment monitoring for the robot arms of the PR2 robot. Sensor data from the laser and stereo sensors on the PR2 robot was fused to generate a consistent model of the environment. The environment model can handle occlusions from parts of the robot's body and also the significant noise in 3-D sensor data. The environment was represented as a mixture of two formats: 1) a voxel grid that represented most of the obstacles in the environment and 2) geometric primitives and mesh models to represent objects that had been recognized and registered in the environment by object detection routines. The environment model was built upon the Octomap package [1], an octree-based probabilistic representation, allowing for efficient representation of large cluttered scenes. The environment model served as the primary input to fast motion planners that could quickly generate collision-free motion plans in fairly cluttered environments. A generic ROS interface for geometric planning allowed multiple motion planners to be integrated, including randomized planners from the Open Motion Planning Library (OMPL) [2], search-based planners from the Search-Based Planning Library (SBPL) [3], and trajectory optimization libraries including CHOMP [4] and stochastic trajectory optimization for motion planning (STOMP) [5]. Some of the motion planners were capable of dealing with geometric constraints, e.g., an orientation constraint specifying that a glass of water should stay upright. Path smoothers and trajectory generators were used to parameterize the planned paths so that smooth trajectories could be executed on the robot. A custom analytical kinematics solver for the PR2 allowed for fast inverse kinematics solutions. An execution component called *move_arm* formed the primary interface to this set of capabilities, exposing an interface that allowed users to specify Cartesian and joint space goals that the robot needed to reach. A trajectory monitoring component kept an eye on the motion of the robot, actively servoing the sensors if necessary, to track the motion of the arm, and respond to changes in the environment. The arm_navigation software was a key component of the grasping pipeline (subject of a previous "ROS Topics" column), allowing the generation of collision-free trajectories to execute, pick, and place actions, obeying the geometric constraints imposed by the task.

Evolution

The evolution of the arm_navigation software to MoveIt! is the result of an effort to package the core algorithmic capabilities in the arm_navigation software separately from the middleware (ROS), allowing for easier code reuse. Lightweight ROS bindings and wrappers will allow ROS users to easily configure and interface with the components of MoveIt!. Advanced users and application developers will be able to directly incorporate the core capabilities in MoveIt! through libraries without needing to depend on a large part of ROS. MoveIt! builds on new features in ROS, including the ability to use the ROS message generation infrastructure as a standalone capability. MoveIt! also integrates the Fast Collision Library (FCL) [6], which provides distance-checking capabilities that should allow for the

Digital Object Identifier 10.1109/MRA.2011.2181749
Date of publication: 26 March 2012



A snapshot of the PR2 robot using the motion planning capabilities in ROS to execute a constrained motion plan as part of a pick-and-place task while keeping the object (a stapler) horizontal. The visualization to the right shows the environment representation, built from onboard sensors, used by the robot for motion planning.

generation of better-quality motion plans. MoveIt! is also expected to contribute in other application areas, including automating the calibration process for robotic arms with 3-D and other visual sensors. The arm_navigation software stack will retain its functionality, but its implementation will consist primarily of arm-specific configurations for the MoveIt! software.

Portability

The arm_navigation software was primarily designed for the PR2 robot but has been used on several other robotic systems. In MoveIt!, we make an additional effort to extract all the needed information into the existing URDF (the standard robot description format in ROS) and newly defined Semantic Robot Description Format (SRDF) to avoid dependencies on specific robots. Furthermore, new tools such as the Arm Navigation Wizard, specifically designed to enable configuration of the arm_navigation software for other robotic manipulators and systems, are being ported to MoveIt!. The Arm Navigation Wizard uses a combination of URDF and SRDF to autoconfigure the arm_navigation software for a robot. An interactive graphical user interface allows users to specify motion-plan requests for their new robots, with a minimal amount of interface implementation required from the user on the robot side. This allows nonmotion planning experts to easily configure the motion planning and associated components in ROS for their own robots.

The Future

We expect MoveIt!, in concert with the motion planning capabilities in OMPL and other packages, to form the basis for new and exciting applications with robots. As MoveIt! evolves and finds more uses in the community, new features and capabilities will find their way into the system, e.g., active sensing, anomalous event detection and response, and the generation of more natural-looking paths. We also aim to develop a set of benchmarking tests around MoveIt!, allowing for the comparison of motion planners and other components for tasks in real-world situations. MoveIt! will be crucial in our goal of enabling safer, more capable robots that can function effectively in human environments.

References

[1] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. (2010). OctoMap: A probabilistic flexible compact 3D map representation for robotic systems. In *Proc. ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation* [Online]. Available: http://octomap.sf.net/

[2] I. A. Sucan, M. Moll, and L. Kavraki. (2010). The Open Motion Planning Library (OMPL), [Online]. Available: http://ompl.kavrakilab.org

[3] B. Cohen and M. Likhachev. (2009). The Search Based Planning Library (SBPL) [Online]. Available: http://www.ros.org/wiki/sbpl

[4] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *Proc. Int. Conf. Robotics and Automation*, May 2009, pp. 489–494.

[5] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *Proc. Int. Conf. Robotics and Automation*, Shanghai, China, May 2011.

[6] J. Pan and D. Manocha. (2011). The Fast Collision Library (FCL) [Online]. Available: http://www.ros.org/wiki/fcl_ros.